

Computación Neuronal y Evolutiva

Informe de prácticas

Práctica 06: Optimización Multi-Objetivo

Alumnos:

1.- Raúl Negro Carpintero

2.- Mario Núñez Izquierdo

Tabla de contenido

[DESCRIPCIÓN 4](#_Toc502323531)

[Restricciones 4](#_Toc502323532)

[Multi-objetivo 5](#_Toc502323533)

[SOLUCIÓN 5](#_Toc502323534)

[Restricciones 5](#_Toc502323535)

[Multi-objetivo 6](#_Toc502323536)

[COMPARATIVA 6](#_Toc502323537)

[Restricciones 6](#_Toc502323538)

[Multi-objetivo 14](#_Toc502323539)

[Ilustración 1 Configuración de los individuos. 4](#_Toc502323540)

[Ilustración 2 Ejemplo de configuración de experimento. 4](#_Toc502323541)

[Ilustración 3 Función de factibilidad. 5](#_Toc502323542)

[Ilustración 4 Función de distancia. 5](#_Toc502323543)

[Ilustración 5 Mejor individuo devuelto por el algoritmo con las restricciones. 6](#_Toc502323544)

[Ilustración 6 Configuración del 1er experimento. 7](#_Toc502323545)

[Ilustración 7 Configuración del 2º experimento. 7](#_Toc502323546)

[Ilustración 8 Configuración del 3er experimento. 7](#_Toc502323547)

[Ilustración 9 Fitness del 1er experimento (1ª ejecución). 7](#_Toc502323548)

[Ilustración 10 Gráfica del 1er experimento (1ª ejecución). 8](#_Toc502323549)

[Ilustración 11 Fitness del 1er experimento (2ª ejecución). 8](#_Toc502323550)

[Ilustración 12 Gráfica del 1er experimento (2ª ejecución). 9](#_Toc502323551)

[Ilustración 13 Fitness del 2º experimento (1ª ejecución). 9](#_Toc502323552)

[Ilustración 14 Gráfica del 2º experimento (1ª ejecución). 10](#_Toc502323553)

[Ilustración 15 Fitness del 2º experimento (2ª ejecución). 10](#_Toc502323554)

[Ilustración 16 Gráfica del 2º experimento (2ª ejecución). 11](#_Toc502323555)

[Ilustración 17 Fitness del 3er experimento (1ª ejecución). 11](#_Toc502323556)

[Ilustración 18 Gráfica del 3er experimento (1ª ejecución). 12](#_Toc502323557)

[Ilustración 19 Fitness del 3er experimento (2ª ejecución). 12](#_Toc502323558)

[Ilustración 20 Gráfica del 3er experimento (2ª ejecución). 13](#_Toc502323559)

[Ilustración 21 Fitness del mejor resultado de la práctica 3. 13](#_Toc502323560)

[Ilustración 22 Gráfica del mejor resultado de la práctica 3. 14](#_Toc502323561)

# DESCRIPCIÓN

*Nota:* todas las pruebas descritas en este documento están realizadas usando el fichero “me\_at\_the\_zoo.csv”.

## Restricciones

En este caso, hemos configurado los individuos y la población en el archivo *principal.py*. Además, en este fichero configuramos los tres experimentos que vamos a evaluar (cruce, mutación, selección y probabilidades), y mostramos los resultados obtenidos mediante gráficas.

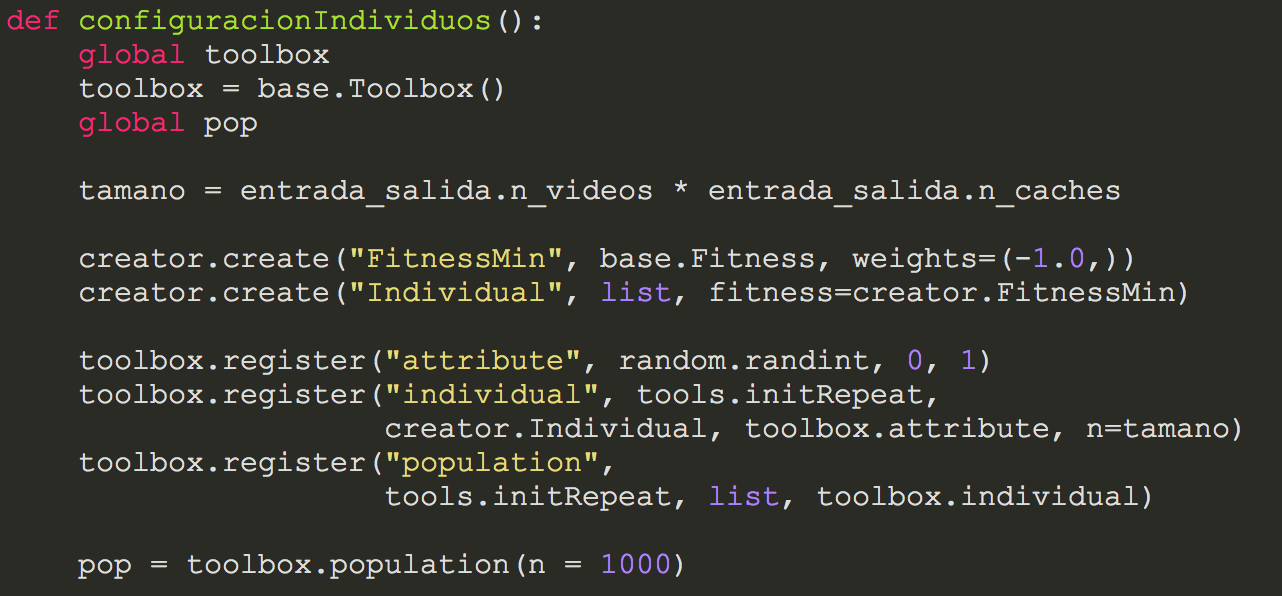


Ilustración Configuración de los individuos.

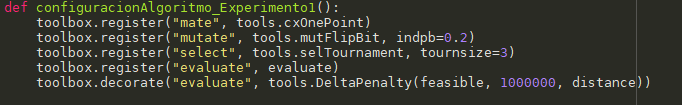


Ilustración Ejemplo de configuración de experimento.

En el fichero *evaluacion.py* implementamos la nueva función de evaluación (prácticamente igual que en las otras prácticas), y las nuevas funciones de factibilidad y distancia para hacer uso de las restricciones.

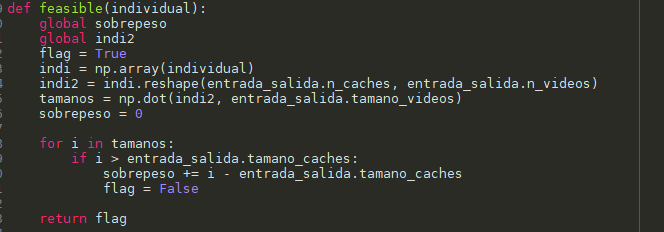


Ilustración Función de factibilidad.

En la función *feasible*, comprobamos si nos hemos pasado de peso en la cache y devolvemos True si no nos hemos pasado (es factible) o False en caso contrario (no es factible).



Ilustración Función de distancia.

En la función *distance*, penalizamos los individuos que no sean factibles.

También hemos hechos cambios menores en el módulo *entrada\_salida.py*, ya que en las anteriores prácticas no cogíamos del todo bien los datos.

## Multi-objetivo

En este caso hemos dividido el programa de la misma manera. En *principal.py* inicializamos los individuos y configuramos los experimentos:

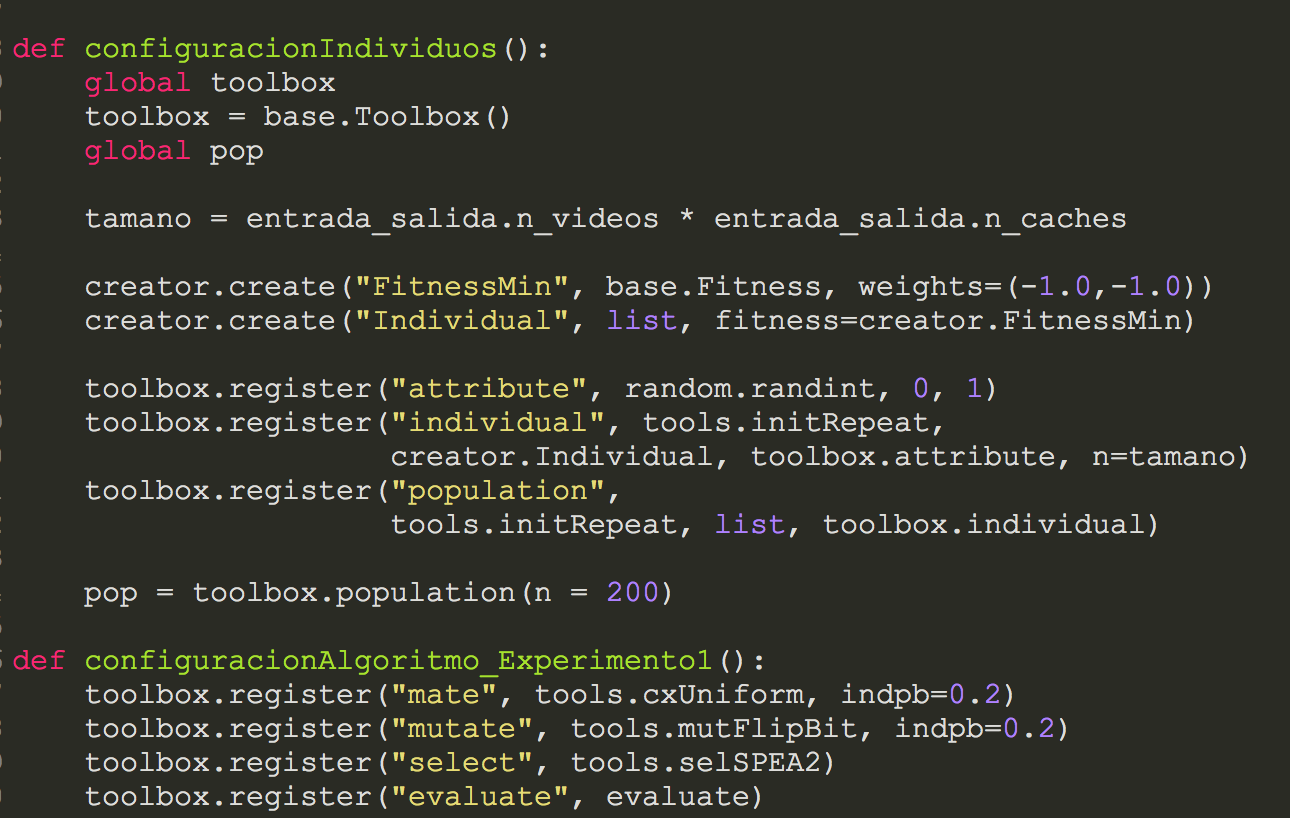


Ilustración Inicialización de individuos y de configuración del multi-objetivo.

Otra cosa a destacar en este fichero (y que lo diferencia del anterior) es que no podemos usar el eaSimple, hemos tenido que utilizar eaMuPlusLambda.



Ilustración Estructura del eaMuPlusLambda.

En *evaluacion.py* implementamos la nueva función de evaluación. Esta función está basada en la que utilizamos en prácticas anteriores, pero con la diferencia de que ahora retorna dos valores en vez de uno.

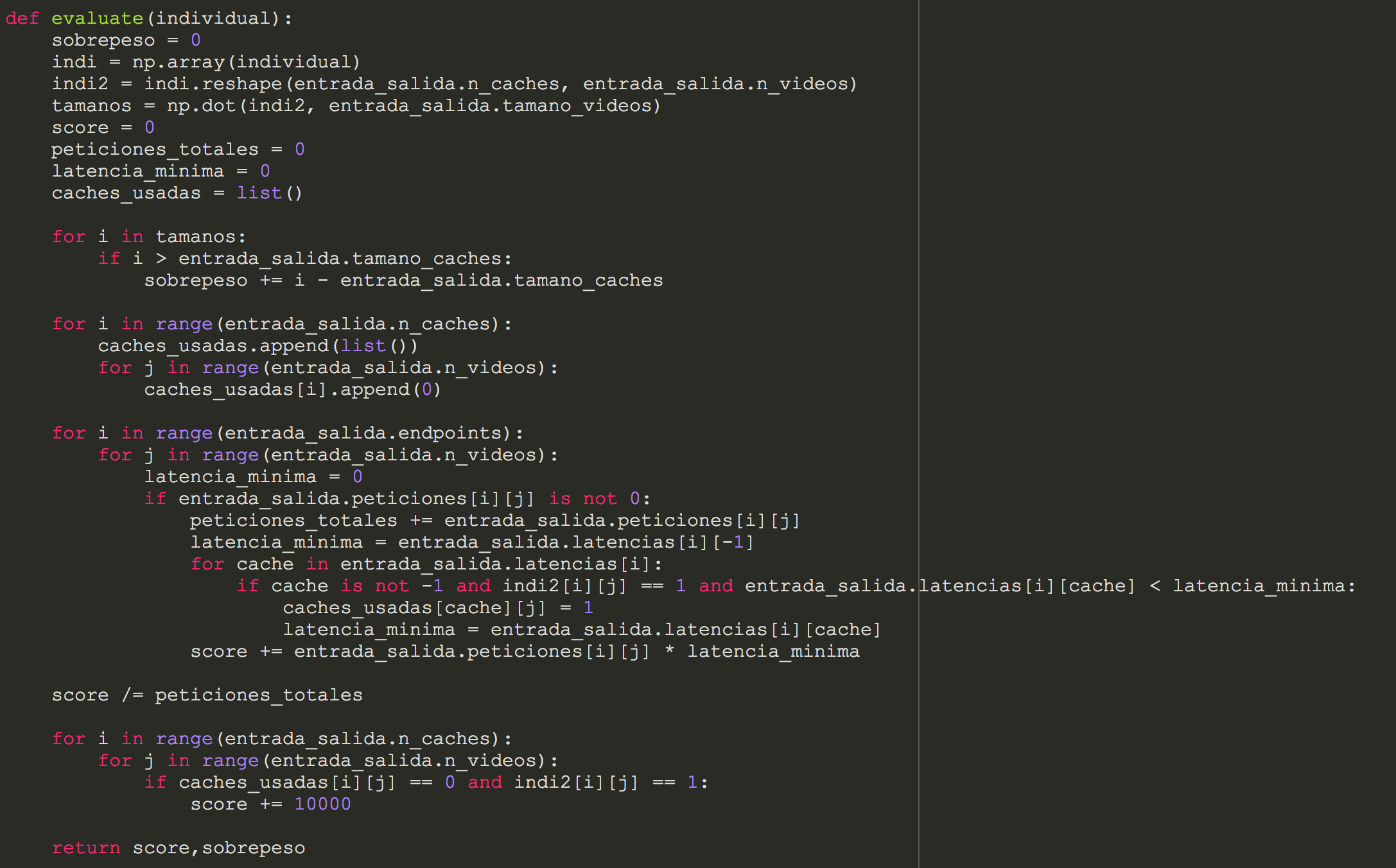


Ilustración Función de evaluación del multi-objetivo.

El módulo *entrada\_salida.py* es el mismo que en el caso anterior.

# COMPARATIVA

## Restricciones

La configuración de los experimentos es la siguiente:

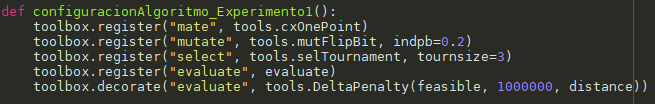


Ilustración Configuración del 1er experimento.

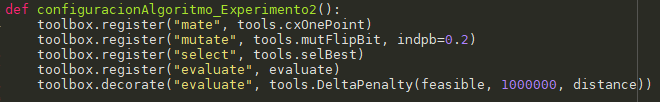


Ilustración Configuración del 2º experimento.

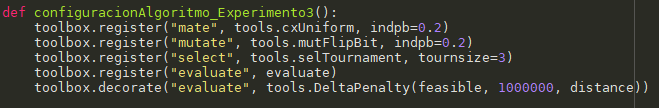


Ilustración Configuración del 3er experimento.

Hemos ejecutado el algoritmo dos veces para comprobar la tendencia de cada experimento, la primera vez ejecutamos con 100 individuos y 100 generaciones y la segunda vez ejecutamos con 1000 individuos y 100 generaciones. Obtuvimos los siguientes resultados:



Ilustración Fitness del 1er experimento (100 individuos 100 gens).

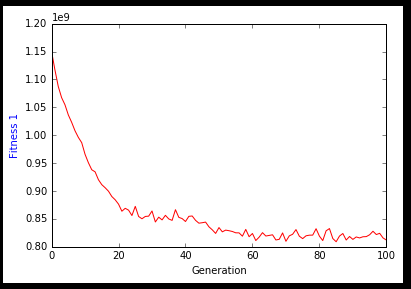


Ilustración Gráfica del 1er experimento (100 individuos 100 gens).



Ilustración Fitness del 1er experimento (1000 individuos 100 gens).

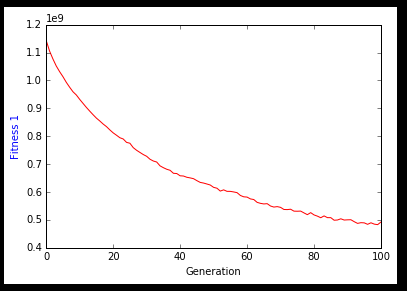


Ilustración Gráfica del 1er experimento (1000 individuos 100 gens).

Como podemos observar, el número de individuos repercute severamente en el resultado, pero aun con todo, la solución que obtenemos es bastante mala y corresponde a un individuo no apto (no respeta el tamaño de las caches).



Ilustración Fitness del 2º experimento (100 individuos 100 gens).

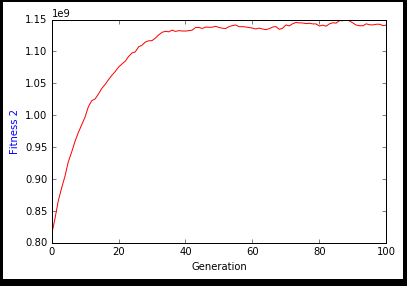


Ilustración Gráfica del 2º experimento (100 individuos 100 gens).



Ilustración Fitness del 2º experimento (1000 individuos 100 gens).

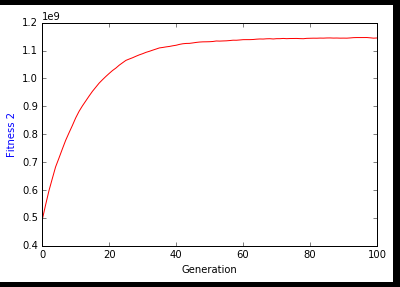


Ilustración Gráfica del 2º experimento (1000 individuos 100 gens).

En este caso, el número de individuos no ha cambiado mucho el resultado. Además, este método de selección no esta funcionando correctamente ya que el fitness esta aumentando y nosotros buscamos disminuirlo.

El único parámetro distinto entre los dos experimentos es el método de selección, siendo selección por torneo en el primero y selección del mejor en el segundo, por lo que podemos asegurar que éste es el causante de las diferencias en los resultados.



Ilustración Fitness del 3er experimento (100 individuos 100 gens).

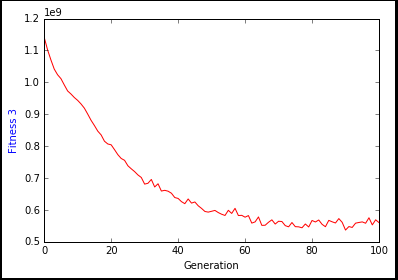


Ilustración Gráfica del 3er experimento (100 individuos 100 gens).



Ilustración Fitness del 3er experimento (1000 individuos 100 gens).

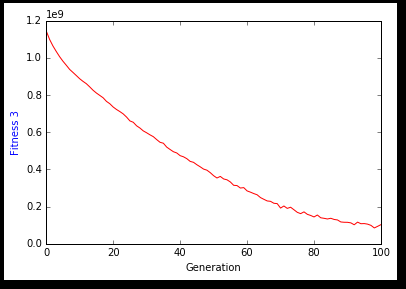


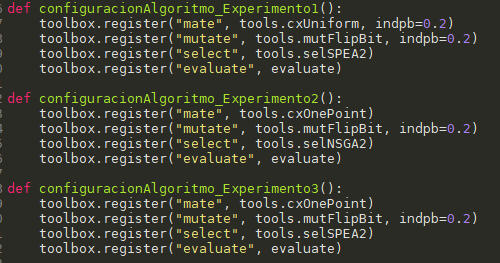
Ilustración Gráfica del 3er experimento (1000 individuos 100 gens).

En este caso, hemos conseguido obtener una solución válida, 860564. Como hemos visto en la configuración de los algoritmos, como mínimo, un individuo no apto tiene 1 millón de fitness por defecto y este valor se ve incrementado aún más por la función distance. Como el fitness obtenido es menor que 1 millón, deducimos que este fitness pertenece a un individuo valido. Sin embargo, no parece ser la más óptima ya que la gráfica no llega a estabilizarse, esto es debido a que 100 generaciones no son suficientes para hallar una solución óptima.

Con estos resultados, podemos deducir que el peor método de selección es el del mejor, y que hay una mejora importante entre el cruce de un punto y el cruce uniforme (el mejor es el uniforme).

## Multi-objetivo

De forma similar al algoritmo anterior, tenemos 3 experimentos preparados y los ejecutaremos 2 veces cada uno. La primera con 100 individuos y 100 generaciones y la segunda con 1000 individuos y 100 generaciones.



La configuración del eaMuPlusLambda es la siguientes para ambas ejecuciones:



Ilustración Configuracion eaMuPlusLambda

Estos son los resultados obtenidos:

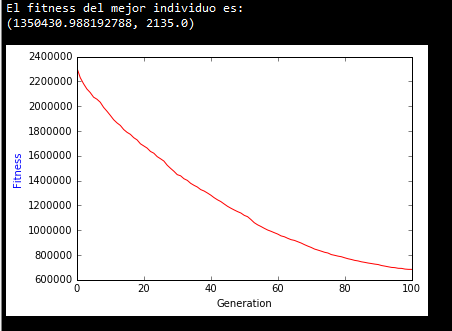


Ilustración Resultados del primer experimento MultiObjetivo (100 individuos 100 gens)

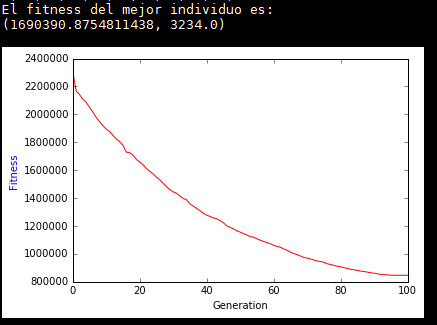


Ilustración Resultados del primer experimento MultiObjetivo (1000 individuos 100 gens)

Como podemos observar, el algoritmo no es capaz de encontrar una solución válida al problema, ya que el segundo fitness nos indica que el mejor individuo obtenido tiene alrededor de 2000 y 3000 megas que sobrepasan el tamaño de las caches.

Además, en este caso nos damos cuenta de que aumentar el número de individuos no ha mejorado la solución si no que, al contrario, la ha empeorado. Esto se debe principalmente a la configuración del eaMuPlusLambda, en ambos casos está configurado para sustituir solo 80 individuos de la población y generar 150 sucesores. Estos datos parecen ser más adecuados para ejecutar con 100 individuos que con 1000.

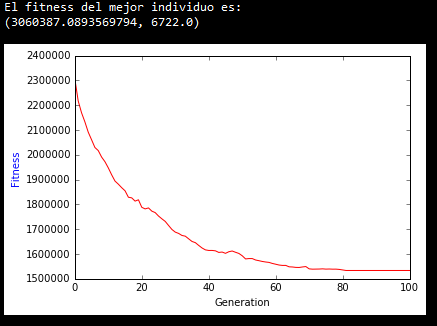


Ilustración Resultados del segundo experimento MultiObjetivo (100 individuos 100 gens)

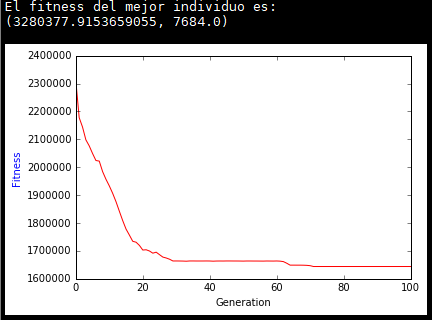


Ilustración Resultados del segundo experimento MultiObjetivo (1000 individuos 100 gens)

En este segundo experimento vemos que los resultados son peores que en el anterior. No obstante, el método de selección nuevo cumple con su función y disminuye el fitness. Y una vez más, en este caso el aumento del número de individuos también ha afectado negativamente a la solución.

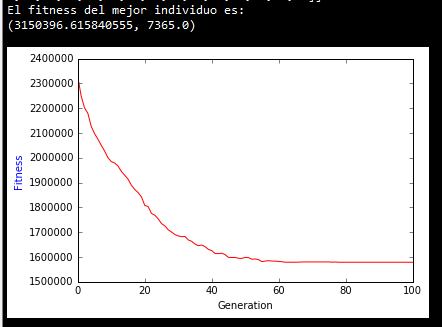


Ilustración Resultados del tercer experimento MultiObjetivo (100 individuos 100 gens)

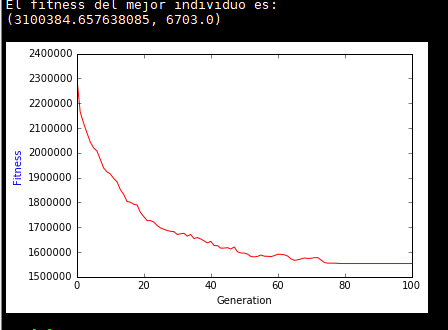


Ilustración Resultados del tercer experimento MultiObjetivo (1000 individuos 100 gens)

En el último de los experimentos volvemos a encontrar unos valores inferiores a los obtenidos en el primero. Lo destacable de este experimento, es que el aumento de individuos ha mejorado la solución, al contrario que en los otros dos.

# MEJOR SOLUCIÓN

## Restricciones

El mejor individuo devuelto por el algoritmo es el siguiente:

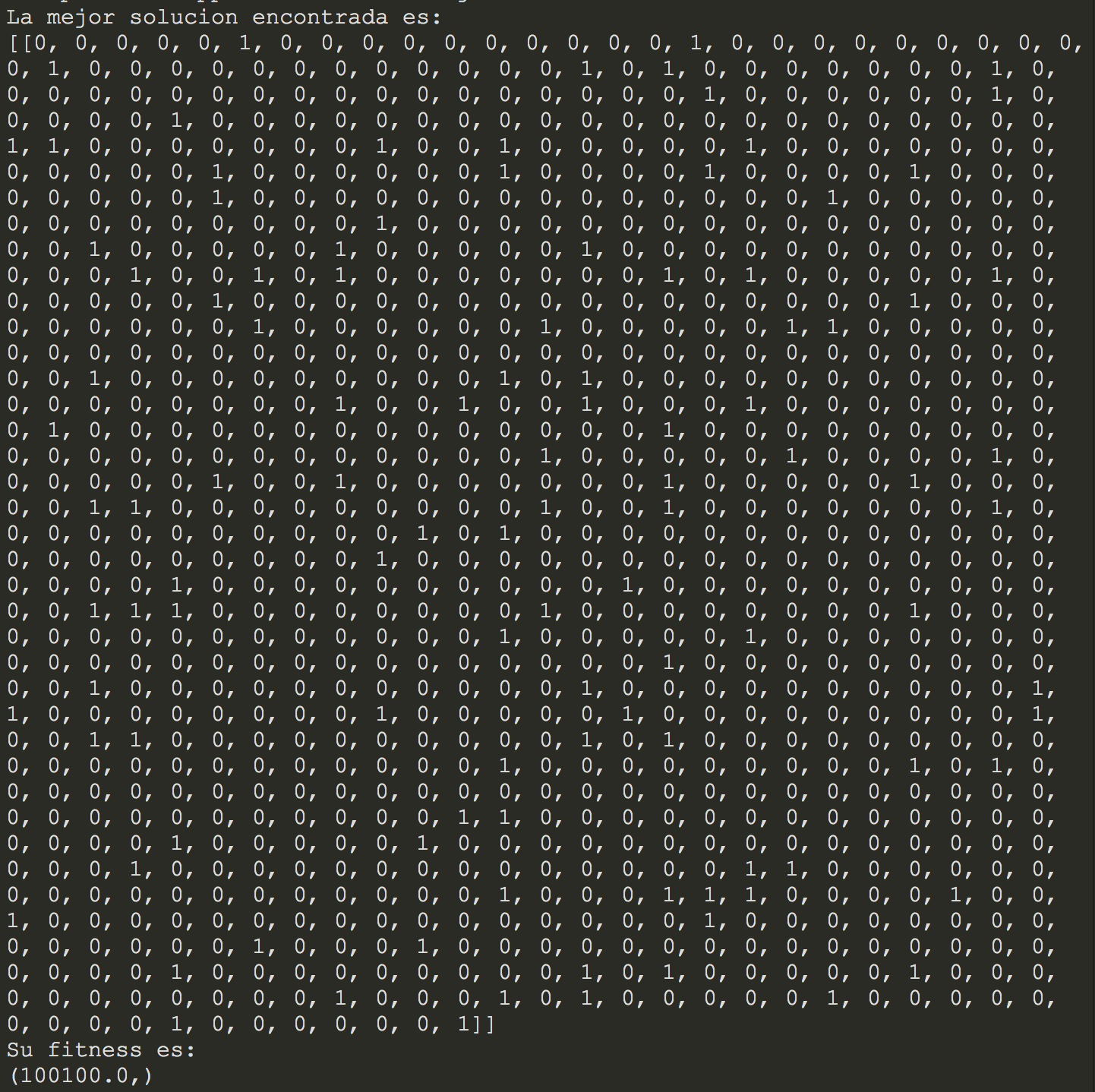


Ilustración 8 Mejor individuo devuelto por el algoritmo con las restricciones.

PONER ALGO MÁS???????????????????????FDSJKBFJKSBGHJDFBGHJDF

## Multi-objetivo

VAYA MOVIDA TÚ

# CONCLUSIÓN

## Restricciones

Si comparamos el mejor resultado de esta práctica con el mejor resultado de la práctica 3, podemos observar que los resultados han mejorado mucho.



Ilustración 33 Fitness del 3er experimento (1000 individuos 100 gens).

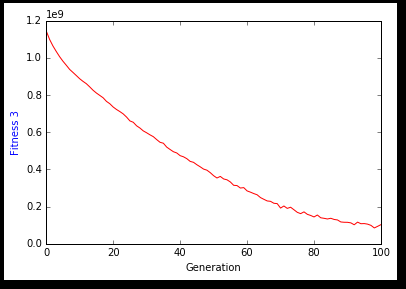


Ilustración 34 Gráfica del 3er experimento (1000 individuos 100 gens).



Ilustración 35 Fitness del mejor resultado de la práctica 3.

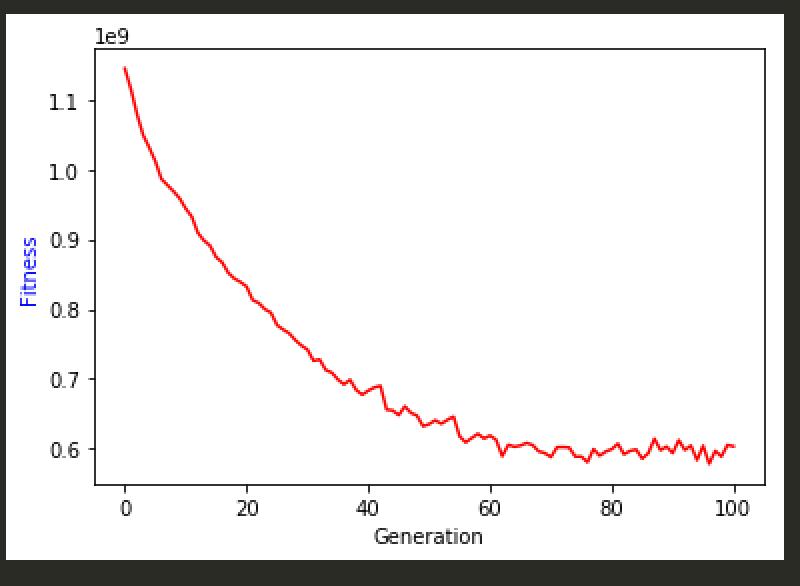


Ilustración 36 Gráfica del mejor resultado de la práctica 3.

Además, hemos observado que la configuración de la practica 3 y la del mejor resultado de esta comparativa son las mismas (cxUniform, mutFlipBit y selTournament).

## Multi objetivo

Estas es la mejor solución obtenida en los tres experimentos (pertenece al primero). De una forma similar al problema de las restricciones, la configuración ganadora es: cxUniform, mutFlipBit y selSPEA2. La forma de selección debe cambiar estrictamente, pero los métodos de cruce y mutación permanecen iguales. Y de forma similar también al problema anterior, parece que aumentando el número de generaciones la solución podría seguir mejorando.

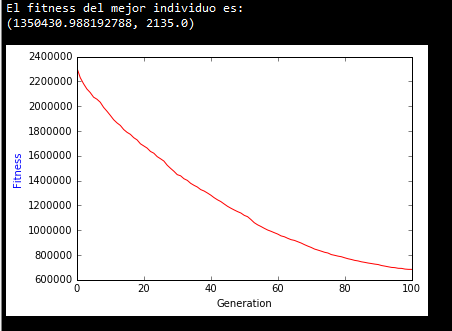


Ilustración 37 Resultados del primer experimento MultiObjetivo (100 individuos 100 gens)

Si la comparamos con la obtenida en la practica 3…

COUSACOUSACOUSA